

1a Parte da P2 de Compiladores 2009

Sintaxe em notação de Wirth

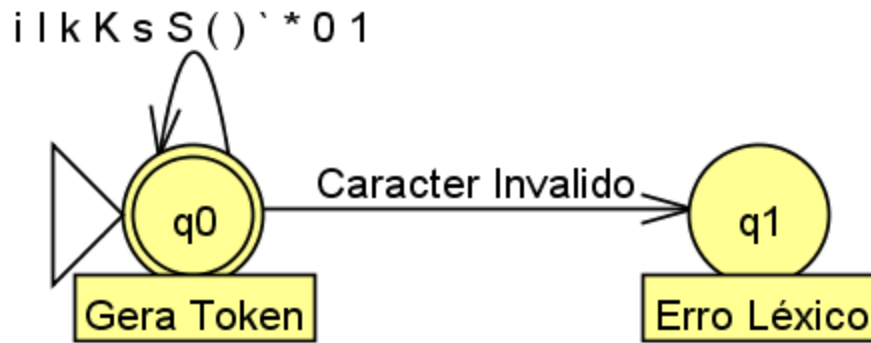
```
Program = {Expr}.  
Expr = ("i" | Expr_).  
IotaExpr = ("i" | Expr_).  
quoteExp = "`" Expr Expr.  
astIota = "*" IotaExpr IotaExpr.  
Expr_ = "I" | "K" | "k" | "S" | "s" | NonemptyJotExpr | quoteExp | IotaExpr | "(" CCExp  
")".  
NonemptyJotExpr = ( "0" | "1" ) { "0" | "1" } .
```

As subMáquinas quoteExp e astIot são criadas "a mais" para simplificar a análise semântica. Não preciso me preocupar em guardar , ou empilhar, a informação semântica "embutida" em "`" ou "*"

Análise Léxica

A análise léxicadesta linguagem é bastante simples pois todos os terminais possuem apenas 1 caractere. A análise léxica, portanto, se resume a ler um caractere e , caso ele seja um dos terminais da linguagem, gerar um token, caso contrário, gerar um erro léxico.

Os terminais deta linguagem são: "i","I","K","k","S","s","`","*","0","1","(",")"



Análise sintática

Cada não terminal da notação de Wirth apresentada se materializa no compilador como uma máquina sintática. A seguir apresentamos as transições criadas para cada máquina. A seguir é apresentado para cada máquina, seu estado inicial, estados finais, transições, identificador de ação semântica e possível ação semântica a executar, bem como o correspondente automato.

As transições que têm como entrada uma máquina, empilham a máquina em execução e o próximo estado e, ao término de execução de uma submáquina, a máquina anterior é desempilhada juntamente com o próximo estado.

Program

Marcação de estados:

Program = 0 { 1 Expr 2 } 1 .

initial: 0

final: 1

(0, Expr) -> 1 Ação semântica: Nenhuma

(1, Expr) -> 1 Ação semântica: Nenhuma



Expr

Marcação de estados:

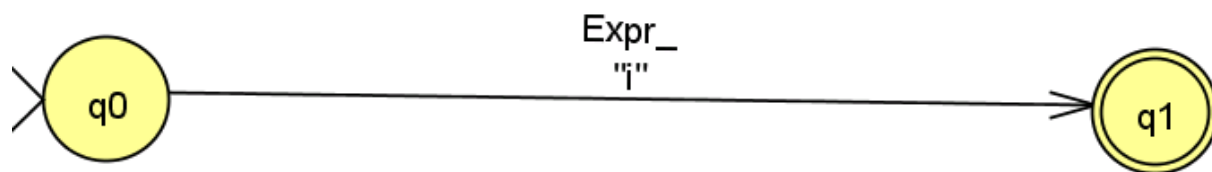
Expr = 0 (0 "i" 2 | 0 **Expr**_ 3) 1 .

initial: 0

final: 1

(0, "i") -> 1 Ação semântica: gera (lambda (x) x)

(0, Expr_) -> 1 Ação semântica: Nenhuma



IotaExpr

Marcação de estados:

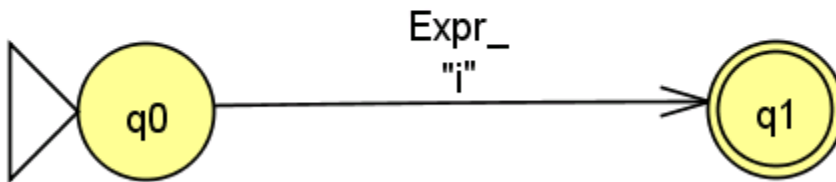
IotaExpr = 0 (0 "i" 2 | 0 Expr_ 3) 1 .

initial: 0

final: 1

(0, "i") -> 1 Ação semântica: gera (lambda (x) (x S K))

(0, Expr_) -> 1 Ação semântica: Nenhuma



quoteExp

Marcação de estados:

quoteExp = 0 "``" 1 Expr 2 Expr 3 .

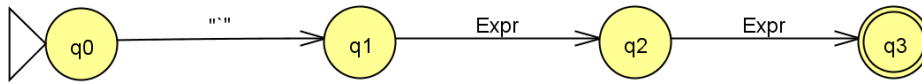
initial: 0

final: 3

(0, "``") -> 1 Ação semântica: Nenhuma

(1, Expr) -> 2 Ação semântica: gera Expr1

(2, Expr) -> 3 Ação semântica: gera Expr2



astIota

Marcação de estados:

astIota = 0 "***" 1 **IotaExpr** 2 **IotaExpr** 3 .

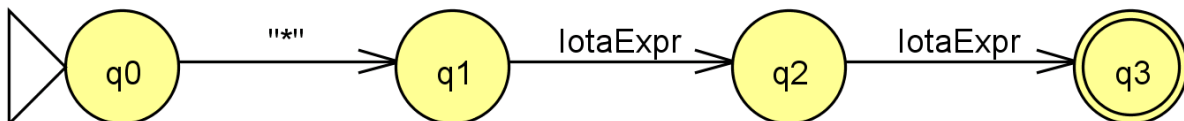
initial: 0

final: 3

(0, "***") -> 1 Ação semântica:

(1, IotaExpr) -> 2 Ação semântica: **gera** IotaExpr1

(2, IotaExpr) -> 3 Ação semântica: **gera** IotaExpr2



Expr_

Marcação de estados:

```
Expr_ = 0 "I" 1 | 0 "K" 2 | 0 "k" 3 | 0 "S" 4 | 0 "s" 5 | 0 NonemptyJotExpr 6 | 0
quoteExp 7 | 0 | 0 "(" 8 CCEExpr 9 ")" 10 .
```

```
initial: 0
```

```
final: 1
```

(0, "I") -> 1 Ação semântica: gera (lambda (x) x)

(0, "K") -> 1 Ação semântica: gera (lambda (x y) x)

(0, "k") -> 1 Ação semântica: gera (lambda (x y) x)

(0, "S") -> 1 Ação semântica: gera (lambda (x y z) ((x z) (y z)))

(0, "s") -> 1 Ação semântica: gera (lambda (x y z) ((x z) (y z)))

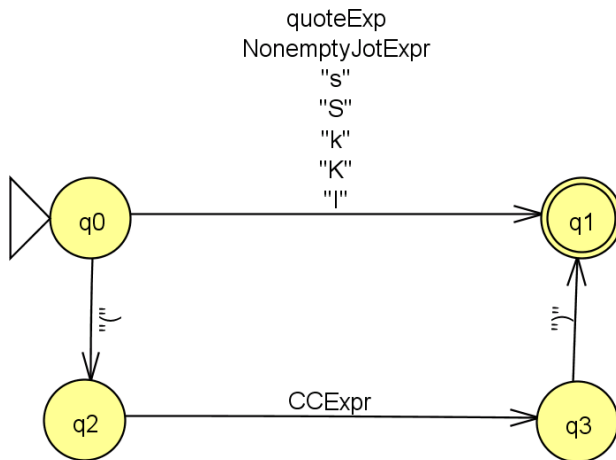
(0, NonemptyJotExpr) -> 1 Ação semântica: Nenhuma

(0, quoteExp) -> 1 Ação semântica: Nenhuma

(0, "(") → 2 Ação semântica: Gera (

(2, CCE Expr) \rightarrow 3 Ação semântica: Nenhuma

(3, " ") -> 1 Ação semântica: Gera)



NonemptyJotExpr

Marcação de estados:

$$\text{NonemptyJotExpr} = 0 \ (0 \ "0" \ 2 \mid 0 \ "1" \ 3) \ 1 \ \{ 4 \ "0" \ 5 \mid 4 \ "1" \ 6 \} \ 4 .$$

```
initial: 0
```

```
final: 1
```

$(0, "0") \rightarrow 1$ Ação semântica:

$(0, "1") \rightarrow 1$ Ação semântica:

$(1, "0") \rightarrow 1$ Ação semântica:

(1, "1") -> 1 Ação semântica:

